

DESIGN CONSIDERATIONS FOR WEB-BASED APPLICATIONS

Luke Wroblewski

University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications

Esa M. Rantanen

University of Illinois at Urbana Champaign, Institute of Aviation

The rapid growth of weblications—application-oriented software delivered as a service over the Web—has revealed a lack of effective guidelines for their design and implementation. Existing Web usability guidelines hinder weblication usability since they are primarily based on interactions within a browsing metaphor. Interface design guidelines for client applications, on the other hand, do not address the conventions of Web users, limitations of the Web environment, nor the new possibilities inherent in the Web. This paper outlines a set of guidelines for weblication interfaces, which fuse translated client-application guidelines and re-purposed Web usability guidelines together to form a solid foundation for weblication interface design. The goal of these recommendations is to address the issues associated with weblication interface design and to present an overall method for thinking about them, emphasizing an understanding of how and why this approach should be adopted.

INTRODUCTION

The onset of the World Wide Web (WWW) began with a massive transfer of information-rich content to the notional realm termed "cyberspace." The generalized interface developed for accessing these stockpiles of information used the "browsing" metaphor, which allowed both a casual inspection of content and specific retrievals through hypertext links and search engines. This interface model was appropriate due to the original nature of the WWW, with its direct correlation to repository of information. However, as the current rise of web-based applications show, services, rather than browsing, are the future (Norman, 2000). These "weblications"—a term first coined by Bruce Tognazzini (Norman, 2000)—are defined as serious software being delivered as a service over the Web (Nail, 1998). The term "serious" is applied because weblications do not include "trivial applets or simple forms" that do not accomplish much beyond collecting information (Berst, 1997). Examples of weblications include Unext.com's Cardean University courses, a system for taking university level classes online, and Currenex's FXTrades system, designed for facilitating online currency exchanges. Such services often include far more complex interactions than simple information retrieval. The browsing metaphor, therefore, is not well suited for weblications and often may hinder weblication usability.

Despite the rapid growth of weblications today, no effective guidelines for their design and implementation exist. Client application graphical user interface (GUI) guidelines do not cover the spectrum of possibilities available on the Web, nor do they take into account the conventions and behavioral patterns that have emerged as the Web has become widely used. Available Web usability guidelines, on the other hand, are not flexible enough to accommodate the new levels of

interaction needed within weblications, and often are not appropriate because a weblication user's motivation differs from a Web site users' goals. The inadequacies of these existing guidelines necessitate the development of new weblication-specific interface design principles.

THESIS

A number of well-developed Web usability guidelines (e.g., Lynch & Horton, 1999; Nielsen, 2000) work well when applied to traditional browsing-oriented web sites. Numerous well-researched GUI guidelines also exist (e.g., Apple Computer, 1992; Galitz, 1996). However, neither set of principles alone is applicable to weblication interface design, since weblications could be thought of as half web site and half client-application. Existing web usability guidelines could hinder weblication usability by restricting interactions to those found in the browsing model. Interface design guidelines for client applications, on the other hand, do not address the WWW, which is at the same time the most gripping and the most limiting factor of weblications. Conventions of Web users, current limitations of the Web environment, and the new possibilities inherent in the Web must be considered during the design process. Client GUI guidelines (the more adaptable of the two sets of guidelines) must be translated into Web terms and supplemented with additional guidelines that emerge from the Web environment to fully encompass the needs of weblication interface design.

SOURCES

Norman (2000) has addressed issues distinguishing weblication design from Web design. These include managing time and workflow within a weblication. Berst

(1997a, 1997b) and Nail (1998) have helped to articulate some of the early concepts behind weblications. Weblication interface design guidelines may be drawn from adapting two existing sets of design principles: GUI designs for client applications and Web usability guidelines for Web sites. Client application interface guidelines (Apple, 1992; Mandel, 1997) articulate GUI design principles as utilized in conventional interfaces, forming the basis for weblication interface guidelines. However, these must be supplemented with Web user conventions and Internet usage issues, which play a vital role in weblications. Web usability guidelines in Nielsen (2000) and Lynch and Horton (1999) outlined principles for designing within the browsing model common to web sites and are important for understanding Web conventions and limitations.

Window use guidelines can be found in Galitz (1996). The problems with windowing solutions are escalated on the Web, however, due to poor browser tools for window management. Weblication interfaces should therefore minimize the use of multiple browser windows. Alternate forms of visual representation of window borders (e.g., Tufte, 1988) are possible, given the weblication development environment. Rollovers (a.k.a. mouseovers) are a form of "just-in-time information delivery" which are highly beneficial for reducing screen clutter while supplying the necessary information when it is needed (Tognazzini, 2000a). This makes rollovers particularly suitable for weblication design. Raskin (2000) provided further rationale behind the principle of indication, which can be utilized as rollovers and alt-overs in weblications. Raskin (2000) also made the case for commonality between application interface techniques, a principle which can be exploited in weblications, and argued against the use of the double-click as an interaction technique. Rationale behind the use of the full screen as a working environment for weblications can be found in Tognazzini (1992), Norman (2000), and Tognazzini's (2000b). Interface innovations have been addressed by Constantine and Lockwood (2000).

FINDINGS

General Design Considerations

Proper weblication interface design comes from an understanding of what makes weblications distinct from standard Web sites and traditional client-applications. Effective solutions to weblication interface issues arise from the criteria, goals, constraints and trade-offs inherent in the nature of weblications.

Weblications are distinct from standard Web sites in many ways. They are not only more interactive, requiring constant user action and reaction, but often times they consist of more complex interactions than those found in traditional Web sites. In addition, these interactions are often unique. Even weblications for similar activities exhibit striking differences in design and workflow. For example, one corporation's online trading system may be drastically different from a competitor's. On the other hand, following

hyperlinks and searching for content on the two corporations' Web sites is likely to be very similar. Also, it is often the case that weblications are more likely to be used more intensively and perhaps more frequently than traditional web pages. Users want the services that weblications have to offer (Norman, 2000). Therefore, they are willing to invest more time in learning the functionality of a weblication for the payoff of increased productivity. This is very much unlike the rapid surfing of Web pages for information, in which case users will promptly move on if they cannot quickly understand the site (Nielsen, 2000).

It is this rapid surfing and informational retrieval that Web usability guidelines aim to optimize. The assumption in Web usability guidelines is that attention is a serious limiting factor for effective interactions. As just pointed out, this is not the case in weblications and hence, it is a source of substantial discrepancies between Web usability guidelines and weblication guidelines.

Client application GUI guidelines are more applicable to weblications, since they are application-orientated and not constrained by the requirements of the browsing model. However, the constraints of the WWW are still very real and need to be taken into account when designing an application to run in the browser environment instead of the desktop environment. Because a weblication is accessed over the Internet, users will perceive its use as similar to that of a Web page, and transfer both applicable and inapplicable knowledge to the weblication. In addition, weblications are subject to be treated in the informal manner common to Web page use. For these reasons, the conventions of Web users need to be taken into careful consideration.

The current limitations of the WWW also constitute an important set of considerations for weblications. Interacting with data over the Internet poses unique problems for users. Delays in download time and connection errors need special accommodations. During delays, users may multitask and need to be reminded of their status within a weblication upon their return. Also, the presentation and development environment of the web is much more limiting than that of the desktop environment. As a result, many interface elements need to be adjusted to work within this new environment. An advantage that results from this common environment is that users can assume the availability of shared commands native to the browser. This unification by domain is important to keep in mind during the design process.

Weblications most often do not occur independent of a Web presence. In order to achieve a unified user experience, a sense of "place" needs to be maintained within a Web site. Therefore, the design and visual presentation of a weblication that belongs in that "place" needs to be seamlessly integrated with the remainder of a Web presence. This is a constraint not often encountered in client application guidelines.

Specific Design Guidelines

These guidelines are derived from research cited above as well as from experience working on the design and development of numerous Web-based applications. This

works encompasses academic as well as corporate applications reviewed by focus groups and informally administered user analysis. Due to a current lack of empirical data to support these recommendations, it is important to point out the goal of this presentation. The objective of these recommendations is to address the issues associated with weblication interface design and to present an overall method for thinking about them. An understanding of how and why to adopt this approach to weblication interface problems is more important than a strict adherence to all the points outlined below. The recommendations should not be interpreted as established guidelines. Rather, their use or disuse in a weblication interface should be supplemented with user testing. Because most weblications are unique, their particular interface solutions are often unique as well.

1. Open the browser window in which weblication interaction occurs to full screen size. Full screen maximizes the information communication capabilities of the display, allows the weblication to be the focus of users' attention, and hides the navigation tools of the web browser window in the infrastructure. The elimination of the web browser tools, borders, and menus makes the browser less confusing when reoriented toward delivering a real application, allowing users to utilize the interaction models of the weblication instead of the browser. Caveats: Inexperienced users may not expect such actions from a browser. Users may also be confused about how to access other applications necessary to complement their interaction with the weblication. It is possible that use of animation could help to clarify the browser's actions.

2. Minimize the use of windows. A minimum number of windows reduces the mental load of managing multiple windows, reduces the possibility for browser windows getting lost behind others, and the confusability of multiple windows. Caveat: Advantages of multi-window systems, such as working on several tasks at once no longer apply. Multiple windows could, however, be contained within a single browser window using Dynamic HTML technologies.

3. Make the windows' content visually dominate window borders. Emphasizing the content of the window over its appearance makes windows less obtrusive. Caveat: Unfamiliar representations of windows may not be perceived as windows, but rather as HTML tables or other visual elements. As a result it may not be obvious to a user that they may be manipulated.

4. Use constant values for fonts, tables, and other visual elements. A consistent layout across browsers and platforms will enhance usability within applications by maintaining a stable interface that users can rely on. Caveat: Separate layouts or accommodating scripts may be needed for the variety of displays and resolutions utilized by web users.

5. Use rollovers. Rollovers are commonly used on the web and users are familiar with them. They also reduce screen

clutter by revealing alternate choices or presenting additional information as it is needed and avoid user confusion by not presenting too many options at once. Caveats: Too many rollovers may result in a flickering effect by constantly having visual items appear and disappear. Some choices and functions will also be unknown to a user until they mouse-over.

6. Use ALT-overs when the immediacy of a rollover is not needed. ALT-overs can clarify the use or function of interface elements without necessitating excessive explanatory labels. ALT-overs are present within many client applications and therefore already familiar to users. Caveat: ALT-overs should not be counted on as the sole clarifier for interface elements.

7. Avoid double clicks. Web users are accustomed to a single click when interacting with web sites. Eliminating double clicks therefore caters to the established conventions of the WWW. Also, double-clicking is problematic for users. The use of a double click requires users to remember which class of interfaced elements responds to a double click and then also what the outcome of the double click will be as opposed to a single click (Raskin, 2000). Caveat: Interface elements within a weblication may visually resemble elements within client applications that require a double-click to activate.

8. Use the conventions of link selection in web. Web users are used to exploration and will move their mouse until a clickable area appears. The users' willingness to explore can facilitate unique forms of interaction. Caveat: This notion should not be stretched too far. Developers should not imbed important functions within hard to find or irrelevant elements.

9. Use inherent functionality of visual elements. Text can initiate actions and images can correspond to various actions as well. Web users have come to expect this from their interfaces. Caveat: Do not imbed important functions within hard to find or irrelevant elements.

10. Use both saturated and unsaturated link colors. A natural mapping exists between executed actions and those not yet executed when saturated and unsaturated versions of a color are used in their visual representation. Caveat: The differences between the saturated and unsaturated versions of a link should not be too subtle.

11. Use underlined fonts as hot-spots. Actions can be embedded within text-based explanations and executed as part of a natural sequence, allowing the information to be adjacent in space to where its use is required. Caveat: Do not imbed important functions within hard to find or irrelevant text elements.

12. Use pull-down menus, radio buttons, and checkboxes as utilized online. Web users are familiar with the functionality of these elements, they are easy to implement, and help to conserve screen space by allowing many possible

options to reside in a small area. Caveat: Avoid non-standard implementation and do not rely on them solely, especially if they are not the proper solution.

13. Use plug-ins and frames as tools for weblication design. Plug-ins increase functionality and alleviate shortcomings of web browsers in the areas of three-dimensional, animated, and sound-based elements while allowing for richer forms of interaction or information display. Caveats: Users are subject to lengthy download times and may not wish to install or utilize a plug-in. The developer must also rely on the plug-in provider for installation and availability of the plug-in.

14. Use motion cues and animation as a feedback mechanism. Animation can be effectively utilized to show continuity and dimensionality in transitions, illustrate change over time, multiplex the display, enrich graphical representations, visualize three-dimensional structures, and attract attention (Nielsen, 2000). Caveats: Poorly implemented animation may distract users. The interface should not depend solely on motion cues for understanding.

15. Use common functionality between a weblication and web browsers. Utilizing interactions common to web browsers allows transfer of knowledge between weblications. Caveat: The interactions available in browsers might not be a good solution for the task at hand.

16. Exploit the similarities in the basic functionality of all weblications. Standardization allows for transfer of knowledge between weblications and eliminates problems associated with implementing non-standard interactions. Caveat: Widespread use of poorly designed interfaces can encourage bad practices among users.

17. Utilize the resource potential of the web in the design of a weblication. Weblication content may be generated by users and documentation can be easily and continually updated in one place and not to be distributed to all users. Caveats: Web may not be the best design platform for an application and a client application might yield a more successful product. Monitoring the users' additions to determine appropriate content is necessary.

18. Manage time and workflow within a weblication. The resulting product will be more task-oriented, guide users through the actions necessary for task completion, and distinguish weblications from content-based web sites. Caveat: Web site developers may lack the skills necessary to develop effective workflow scenarios.

19. Consider the aesthetic integrity of the interface. Aesthetic presentation can give a weblication a personality, provide users enjoyment or familiarity, and a sense of trust and professionalism. Consistent aesthetics unite various sections of a weblication and give it a coherent look and feel. Caveat: Placing too much emphasis on aesthetics can cause

usability to suffer. It is important to not allow visual treatments to overwhelm interaction elements.

DISCUSSION

Ideally, a user should not need to know whether they are interacting with an application over the Internet or on their home computer. Until this can be done seamlessly, we need to make what is possible as effective as possible. Too often an interface that works in one medium is simply translated to a new medium, without regard to whether its usability in the original medium translates to the new "habitat." This scenario has been rehashed time after time in the interface design discipline. It is obvious that the "reapplication" methodology is flawed. We are currently in an interface "rut." The XEROX Parc and Macintosh Lisa projects moved interface design into the graphical realm, eliminating the secretive command line interfaces that were their predecessors. However, despite many advances in personal computers the GUIs in widespread use have changed very little since then. The power of processors, increased memory capacities, and superior graphics handling present in modern computers have the potential to greatly enhance the usability of GUI systems. The cumbersome windowing systems of today can be replaced by highly visual and dynamic environments, which allow users to move through and understand their working environments in new ways. Having learned the "reapplication" lesson, we should do better this time. The majority of services to be available online are still under development or consideration. The time is right to get it right.

There are still considerable obstacles to overcome before universal and widely applicable guidelines can be developed, however. The foremost difficulty arises from the extensive diversity of weblications. Application-specific constraints and considerations could easily override general recommendations and guidelines. For the same reason, empirical evaluation of the aforementioned guidelines is difficult. Laboratory experiments are frequently limited to address only a small subset of all possible variables present in the operational environment and hence, the results may not be readily generalizable.

An alternative to a usability laboratory, and the approach we are considering for further research into the issue, is to collect usability data online from the weblications. This data would allow for analysis of interaction patterns, under-, over-, and misuse of features, and detection of both errors and innovations by the users, which would further guide the development of weblication interfaces. The benefits offered by such data are likely to outweigh the associated drawbacks, including loss of experimental control and handling of massive amounts of data. However, this approach will necessitate the development of novel usability metrics and performance measures, data collection methodologies, and data analysis tools. Participant consent and privacy issues may present novel problems as well, that must be resolved first.

REFERENCES

- Apple Computer, Inc. (1992). *Macintosh Human Interface Guidelines*. Reading, MA: Addison-Wesley Publishing Co.
- Berst, J. (1997a). *How Weblications Are Changing the Internet*. ZDNet Anchor Desk
http://www.zdnet.com/anchordesk/story/story_1020.html
- Berst, J. (1997b). The pros and cons of Weblications. *PC Week Online*.
<http://www.zdnet.com/eweek/opinion/0707/07berst.html>
- Constantine, L L., & Lockwood L. A. D. (2000). *Inventing Interfaces: Tactics, Tricks, and Techniques for Breakthrough Innovations*. North Andover, MA: User Interface Engineering
- Norman, D. A. (2000). The Rise of Weblications: Keynote address presented at the User Experience World Tour. November 17, 2000, Chicago. IL.
- Galitz, W. O. (1996). *The Essential Guide to User Interface Design*. New York. John Wiley & Sons
- Lynch, P. J., & Horton, S. (1999). *Web Style Guide : Basic Design Principles for Creating Web Sites*. New Haven: Yale University Press.
- Mandel, T. (1997). *The Elements of User Interface Design*. New York: John Wiley & Sons.
- Nail, J. (1998). Why did we choose the 'weblication' route. *ZDNet Anchor Desk*,
http://www.zdnet.com/anchordesk/talkback/talkback_93309.html
- Nielsen, J. (2000). *Designing Web Usability: The Practice of Simplicity*. Indianapolis: New Riders Publishing.
- Nielsen, J. (1997a). The Difference Between Web Design and GUI Design. *Jakob Nielsen's Alertbox for May 1, 1997*,
<http://www.useit.com/alertbox/9705a.html>
- Raskin, J.. (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. Reading, MA: Addison-Wesley Publishing Co.
- Tognazzini, B. (2000a) OS X: A First Look. *AskTog*,
<http://www.asktog.com/columns/034OSX-FirstLook.html>
- Tognazzini, B. (2000b) Maximizing Windows. *AskTog*,
<http://www.asktog.com/columns/000maxscrns.html>
- Tognazzini, B. (1992). *Tog on Interface*. Reading, MA: Addison-Wesley Publishing Co.
- Tufte, E. (1989). *Visual Design of the User Interface*. Armonk, NY: IBM Corporation